CURSO DE ELETRÔNICA DIGITAL

INTRODUÇÃO

Os circuitos equipados com processadores, cada vez mais, estão fazendo parte do cotidiano do técnico e/ou engenheiro, tanto de campo como de desenvolvimento.

Hoje, dificilmente encontramos um equipamento, seja ele de consumo ou de produção, que não possua pelo menos um processador (DSP, microprocessador, ou microcontrolador).

É fato também que vários profissionais encontram muitas dificuldades na programação e desenvolvimento de projetos com esses componentes, simplesmente por terem esquecido alguns conceitos fundamentais da eletrônica digital clássica.

A intenção desse “especial” é justamente essa, ou seja, cobrir possíveis lacunas sobre essa tecnologia de modo simples e objetivo. Procuramos complementar a teoria com circuitos práticos e úteis, e dividimos o trabalho em doze capítulos:

- Sistemas de numeração
- Álgebra de Boole e portas lógicas
- Família TTL
- Família CMOS
- Funções lógicas
- Flip-Flops
- Funções lógicas integradas
- Multivibradores
- Contadores
- Decodificadores
- Registradores de deslocamento
- Displays

Tivemos o cuidado de elaborar alguns testes, para que o leitor possa acompanhar melhor sua percepção.

Newton C. Braga

LIÇÃO 1

ELETRÔNICA ANALÓGICA E DIGITAL
SISTEMAS DE NUMERAÇÃO

1.1- ANALÓGICO E DIGITAL

Por que digital? Esta é certamente a primeira pergunta que qualquer leitor que está "chegando agora" e tem apenas alguma base teórica sobre Eletrônica faria ao encontrar o nosso curso.

Por este motivo, começamos justamente por explicar as diferenças entre as duas eletrônicas, de modo que elas fiquem bem claras. Devemos lembrar que em muitos equipamentos, mesmo classificados como analógicos ou digitais, encontraremos os dois tipos de circuitos. É o caso dos computadores, que mesmo sendo classificados como "máquinas estritamente digitais" podem ter em alguns pontos de seus circuitos configurações analógicas.

Uma definição encontrada nos livros especializados atribui o nome de Eletrônica Digital aos circuitos que operam com quantidades que só podem ser incrementadas ou decrementadas em passos finitos.

Um exemplo disso é dado pelos circuitos que operam com impulsos. Só podemos ter números inteiros de pulsos sendo trabalhados em qualquer momento em qualquer ponto do circuito. Em nenhum lugar encontraremos "meio pulso" ou "um quarto de pulso".

A palavra digital também está associada a dígito (do latim digitu, dedo) que está associado à representação de quantidades inteiras. Não podemos usar os dedos para representar meio pulso ou um quarto de pulso.

Na Eletrônica Analógica trabalhamos com quantidades ou sinais que podem ter valores que variam de modo contínuo numa escala. Os valores dos sinais não precisam ser inteiros. Por exemplo, um sinal de áudio, que é analógico, varia suavemente entre dois extremos, enquanto que um sinal digital só pode variar aos saltos, observe a **figura 1**.

Conforme o leitor pode perceber, a diferença básica entre os dois tipos de eletrônica está associada inicialmente ao tipo de sinais com que elas trabalham e no que elas fazem com os sinais.

De uma forma resumida podemos dizer que:

A Eletrônica Digital trabalha com sinais que só podem assumir valores discretos ou inteiros.

A Eletrônica Analógica trabalha com sinais que podem ter qualquer valor entre dois limites.

1.2 - LÓGICA DIGITAL

Os computadores e outros equipamentos que usam circuitos digitais funcionam obedecendo a um tipo de comportamento baseado no que se denomina Lógica.

Diferentemente dos circuitos amplificadores comuns que simplesmente amplificam, atenuam ou realizam algum tipo de processamento simples dos sinais, os circuitos digitais usados em computadores e outras máquinas não processam os sinais baseados em uma finalidade simples determinada quando são fabricados.

Os circuitos digitais dos computadores e outros equipamentos são capazes de combinar os sinais tomando decisões segundo um comportamento lógico.

É evidente que se o leitor deseja realmente entender como as coisas acontecem nos circuitos digitais, deve partir exatamente do aprendizado do comportamento lógico. Podemos dizer que a lógica nos permite tirar

> **COMPUTADORES:** os computadores atuais são digitais em sua totalidade e praticamente não é usado outro tipo de configuração. No entanto, nem sempre foi assim. Nas primeiras décadas deste século, quando os circuitos eram ainda valvulados, os primeiros computadores eram máquinas analógicas. A imprecisão e algumas outras dificuldades técnicas que estes computadores apresentavam fizeram com que logo fossem substituídos pelos circuitos digitais hoje usados.

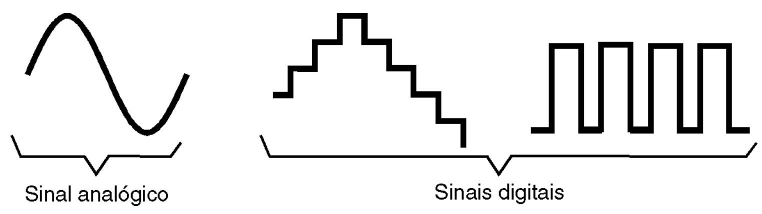

Figura 1 - Os sinais digitais variam aos saltos.

conclusões ou tomar decisões a partir de fatos conhecidos.

Por exemplo, a decisão de “acender uma lâmpada quando está escuro” é uma decisão lógica, pois a proposição e a conclusão são fatos relacionados.

Ao contrário, a decisão de “acender uma lâmpada, porque está chovendo” não é uma decisão lógica, pois os fatos envolvidos não têm relação.

Evidentemente, os fatos relacionados acima são simples e servem para exemplificar como as coisas funcionam.

Na eletrônica dos computadores, o que temos é a aplicação da lógica digital, ou seja, de circuitos que operam tomando decisões em função de coisas que acontecem no seu próprio interior. É claro que os computadores e seus circuitos digitais não podem entender coisas como está escuro ou está chovendo e tomar decisões.

Os circuitos lógicos digitais trabalham com sinais elétricos.

Assim, os circuitos lógicos digitais nada mais fazem do que receber sinais com determinadas características e em função destes tomar decisões que nada mais são do que a produção de um outro sinal elétrico.

Mas, se os sinais elétricos são digitais, ou seja, representam quantidades discretas e se a lógica é baseada em tomada de decisões, o próximo passo no entendimento da Eletrônica Digital, é partir para o modo como as quantidades discretas são representadas e entendidas pelos circuitos eletrônicos.

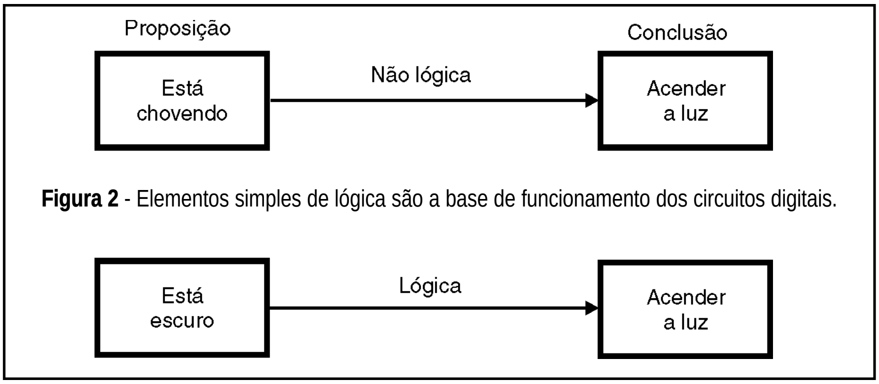

Figura 2 - Elementos simples de lógica são a base de funcionamento dos circuitos digitais.

1.3 - SISTEMAS DE NUMERAÇÃO

O modo como contamos as quantidades vem do fato de possuirmos 10 dedos. Assim, tomando os dedos das mãos podemos contar objetos com facilidade até certo ponto.

O ponto crítico ocorre quando temos quantidades maiores do que 10. O homem resolveu o problema passando a indicar também a quantidade de mãos ou de vezes em que os dez dedos eram usados.

Assim, quando dizemos que temos 27 objetos, o 2 indica que temos “duas mãos cheias” ou duas dezenas mais 7 objetos. O 2 tem peso 10.

Da mesma forma, quando dizemos que temos 237 objetos, o 2 indica que temos “duas dezenas de mãos cheias” ou duas centenas, enquanto o 3 indica que temos mais 3 mãos cheias e finalmente o 7, mais 7 objetos, **figura 3**. Em outras palavras, a posição dos algarismos na representação dos números tem um peso e em nosso sistema de numeração que é decimal este peso é 10, veja a **figura 4**.

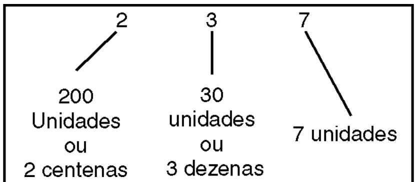

Figura 3 - A posição do algarismo dá seu valor relativo.

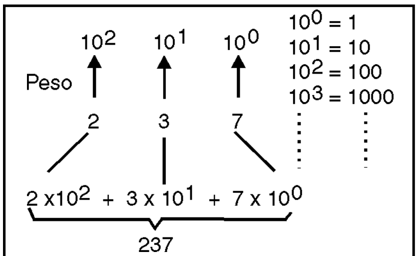

Figura 4 - Os pesos são potências de 10 no sistema decimal.

O que aconteceria se tivéssemos um número diferente de dedos, por exemplo 2 em cada mão?

Isso significaria, em primeiro lugar, que em nosso sistema de base 4 (e não base 10) só existiriam 4 algarismos para representar os números: 0, 1, 2 e 3, confira a **figura 5**.

Para representar uma quantidade maior do que 4 teríamos de usar mais de um algarismo.

Assim, para indicar 7 objetos na base 4, teríamos “uma mão cheia com 4” e mais 3. Isso daria 13, **figura 6**.

Veja então que no “13” na base 4, o 1 tem peso 4, enquanto que o 3 tem o seu valor normal.

De uma forma generalizada, dizemos que dependendo da base do sistema os algarismos têm “pesos” que correspondem à sua posição no número e que estes pesos são potências da base. Por exemplo, para a base 10, cada algarismo a partir da direita tem um peso, que é uma potência de 10 em ordem crescente, o que nos leva à unidade (dez elevado a zero), à dezena (dez elevado ao expoente um), à centena (dez elevado ao quadrado), ao milhar (dez elevado ao cubo) e assim por diante, conforme a **figura 7**.

Em Eletrônica Digital costumamos dizer que o dígito mais à direita, por representar a menor potência ou ter menor peso, é o dígito ou bit* menos significativo ou LSB (*Less Significant Bit*) enquanto que o mais à esquerda é o mais significativo ou MSB (*Most Significant Bit*). Para a base 4, conforme observamos na figura 8, os dígitos têm potências de 4.

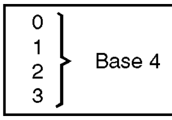

Figura 5 - Na base 4 são usados 4 algarismos.

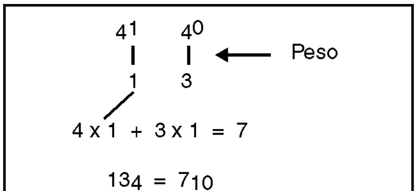

Figura 6 - Treze na base quatro equivale a sete na base 10.

*O bit que é o dígito binário (na base 2) será estudado mais adiante.

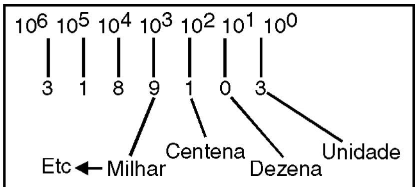

Figura 7 - Os pesos aumentam da direita para a esquerda.

1.4 - NUMERAÇÃO BINÁRIA

Os circuitos eletrônicos não possuem dedos.

É evidente também que não seria muito fácil projetar circuitos capazes de reconhecer 10 níveis de uma tensão ou de outra grandeza elétrica sem o perigo de que qualquer pequeno problema fizesse-os causar qualquer confusão.

Muito mais simples para os circuitos eletrônicos é trabalhar com um sistema de numeração que esteja mais de acordo com o seu princípio de funcionamento e isso realmente é feito.

Um circuito eletrônico pode ter ou não corrente, ter ou não tensão, pode receber ou não um pulso elétrico.

Ora, os circuitos eletrônicos são mais apropriados para operar com sinais que tenham duas condições possíveis, ou seja, que representem dois dígitos ou algarismos.

Também podemos dizer que as regras que regem o funcionamento dos circuitos que operam com apenas duas condições possíveis são muito mais simples.

Assim, o sistema adotado nos circuitos eletrônicos digitais é o sistema binário ou de base 2, onde são usados apenas dois dígitos, correspondentes a duas condições possíveis de um circuito: 0 e 1.

Mas, como podemos representar qualquer quantidade usando apenas dois algarismos?

A idéia básica é a mesma usada na representação de quantidades no sistema decimal: atribuir pesos aos dígitos conforme sua posição no número. Assim, vamos tomar como exemplo o valor 1101 que em binário representa o número 13 decimal e ver como isso ocorre.

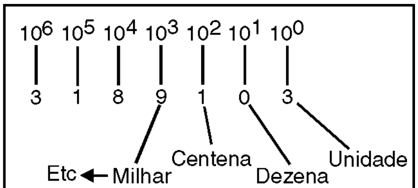

Figura 8 - Os pesos na base 4.

O primeiro dígito da direita nos indica que temos uma vez o peso deste dígito ou 1.

O zero do segundo dígito da direita para a esquerda indica que não temos nada com o peso 2.

Agora o terceiro dígito da direita para a esquerda e que tem peso 4 é 1, o que indica que temos "uma vez quatro".

Finalmente, o primeiro dígito da esquerda que é 1 e está na posição de peso 8, nos diz que temos "uma vez oito".

Somando uma vez oito, com uma vez quatro e uma vez um, temos o total, justamente a quantidade que conhecemos em decimal como treze.

Veja então, conforme indica a **figura 9**, que na numeração binária, os dígitos vão tendo pesos da direita para a esquerda que são potências de 2, ou seja, dois elevado ao expoente zero que é um, dois elevado ao expoente 1 que é 2, dois ao quadrado que é 4 e assim por diante.

Basta lembrar que a cada vez que nos deslocamos para a esquerda, o peso do dígito dobra, **figura 10**.

Como não existe um limite para os valores dos pesos, isso significa que é posível representar qualquer quantidade em binário, por maior que seja, simplesmente usando o número apropriado de dígitos.

Para 4 dígitos podemos representar números até 15; para 8 dígitos podemos ir até 255; para 16 dígitos até 65 535 e assim por diante.

O leitor deve lembrar-se desses valores limites para 4, 8 e 16 dígitos de um número binário, pois eles têm uma grande importância na Informática.

A seguir damos a representação binária dos números decimais até 17 para uma melhor ilustração de como tudo funciona:

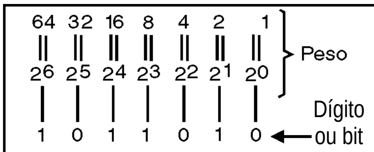

Figura 9 - Pesos na numeração binária.

Decimal	Binário	Decimal	Binário
0	0	9	1001
1	1	10	1010
2	10	11	1011
3	11	12	1100
4	100	13	1101
5	101	14	1110
6	110	15	1111
7	111	16	10000
8	1000	17	10001

Para o leitor que pretende entender de Eletrônica Digital aplicada aos computadores há momentos em que é preciso saber converter uma indicação em binário para o decimal correspondente.

Podemos dar como exemplo o caso de certas placas que são usadas no diagnóstico de computadores e que possuem um conjunto de LEDs que acende indicando um número correspondente a um código de erros. Os LEDs apagados indicam o algarismo 0 e os LEDs acesos, o algarismo 1.

Vamos supor que num diagnóstico a sequência de acendimento dos LEDs seja 1010110. É preciso saber por onde começar a leitura ou seja, se o de menor peso é o da direita ou da esquerda.

Nas indicações dadas por instrumentos ou mesmo na representação da valores binários, como por exemplo na saída de um circuito, é preciso saber qual dos dígitos tem maior peso e qual tem menor peso.

Isso é feito com uma sigla adotada normalmente e que se refere ao dígito, no caso denominado bit.

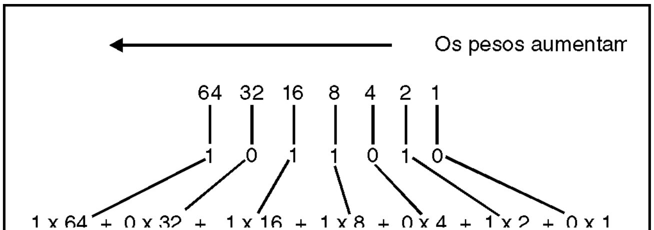

Figura 10 - Na numeração binária os pesos dobram a cada digito deslocado para a esquerda.

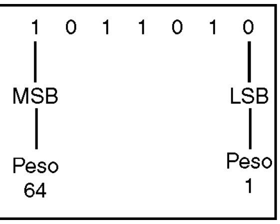

Figura 11 - Extremos de um número binário.

Assim, conforme citado anteriormente, para o dígito de menor peso ou bit menos significativo é adotada a sigla LSB (*Less Significant Bit*) e para o mais significativo é adotada a sigla MSB (*Most Significant Bit*), figura 11.

O que fazemos é somar os valores dados pelos dígitos multiplicados pelo peso de sua posição. No caso do valor tomado como exemplo, 1010110, temos:

Dígito		Peso		Valor
1	x	64	=	64
0	x	32	=	0
1	x	16	=	16
0	x	8	=	0
1	x	4	=	4
1	x	2	=	2
0	x	1	=	0

Somando os valores teremos:
64 + 16 + 4 + 2 = 86

O valor decimal de 1010110 é 86.

Assim, tudo que o leitor tem de fazer é lembrar que a cada dígito que saltamos para a esquerda seu peso dobra na sequência 1, 2, 4, 8, 16, 32, 64, 128, etc.

Na prática também pode ocorrer o problema inverso, transformação de um valor expresso em decimal (base 10) para a base 2 ou binário.

Para esta transformação podemos fazer uso de algoritmo muito simples que memorizado pelo leitor pode ser de grande utilidade, dada sua praticidade.

Para os que não sabem, algoritmo nada mais é do que uma sequência de operações que seguem uma determinada regra e permitem realizar uma operação mais complexa. Quando você soma os números um sobre o outro (da mesma coluna) e passa para cima os dígitos que excedem o 10, fazendo o conhecido "vai um", você nada mais está fazendo do que usar um algoritmo.

Os computadores usam muitos tipos de algoritmos quando fazem suas operações, se bem que a maioria não precise ser conhecida dos leitores.

Assim, para a conversão de um decimal para binário, como por exemplo o 116, o que fazemos é uma série de divisões sucessivas, **figura 12**.

Vamos dividindo os números por 2 até o ponto em que chegamos a um valor menor que 2 e que portanto, não pode mais ser dividido.

O resultado desta última divisão, ou seja, seu quociente é então o primeiro dígito binário do número convertido. Os demais dígitos são obtidos lendo-se os restos da direita para a esquerda da série de divisões que realizamos. Tudo muito simples e rápido.

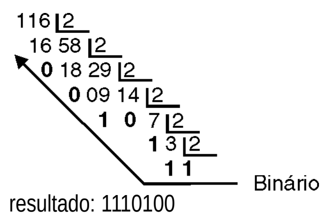

Figura 12 - Conversão de um decimal em binário por divisões sucessivas.

1.5 - BINÁRIOS MENORES QUE 1

Para o leitor talvez seja difícil entender como usando quantidades que só podem ser inteiras, como dado pela definição de digital no início desta lição, seja possível representar quantidades menores que um, ou seja, números "quebrados" ou fracionários.

É claro que isso é possível na prática, pois se assim não fosse os computadores e as calculadoras não poderiam realizar qualquer operação com estes números e sabemos que isso não é verdade.

O que se faz é usar um artifício que consiste em empregar potências negativas de um número inteiro para representar quantidades que não são inteiras.

Assim é possível usar dígitos binários para representar quantidades fracionárias sem problemas.

Vamos dar um exemplo tomando o número 0,01101 em binário.

A própria existência de um "0," já nos sugere que se trata de um número menor que 1 e portanto, fracionário.

Ocorre que os dígitos deste número têm pesos que correspondem a potências de 2 negativas, que nada mais são do que frações, conforme a seguinte sequência:

Dígito		Peso		Valor
0,	x	1	=	0
0	x	1/2	=	0
1	x	1/4	=	0,25
1	x	1/8	=	0,0625
0	x	1/16	=	0
1	x	1/32	=	0,03125

Somando os valores relativos teremos:
0,25 + 0,0625 + 0,03125 = 0,625

O número decimal representado é portanto 0,625.

Veja que usando tantos dígitos quantos sejam necessários podemos representar com a precisão desejada um número decimal.

1.6 - FORMAS DIFERENTES DE UTILIZAR O SISTEMA BINÁRIO

A utilização de circuitos eletrônicos com determinadas características e a própria necessidade de adaptar o sistema binário à representação de valores que sejam convertidos rapidamente para o decimal e mesmo outros sistemas, levou ao aparecimento de algumas formas diferentes de utilização dos binários.

Estas formas são encontradas em diversos tipos de equipamentos digitais, incluindo os computadores.

Sistema BCD (Decimal Codificado em Binário)

BCD é a abreviação de *Binary Coded Decimal* e se adapta melhor aos circuitos digitais.

Permite transformar cada dígito decimal de um número numa representação por quatro dígitos binários (bits) independentemente do valor total do número que será representado.

Assim, partimos da seguinte tabela:

Dígito decimal	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Se quisermos representar em BCD o número 23,25 não o convertemos da forma convencional por divisões sucessivas mas sim, tomamos cada dígito e o convertemos no BCD equivalente, conforme segue:

2 3, 2 5
0010 0011 0010 0101

Veja então que para cada dígito decimal sempre teremos quatro dígitos binários ou bits e que os valores 1010, 1011, 1100, 1101 e 1111 não existem neste código.

Esta representação foi muito interessante quando as calculadoras se tornaram populares, pois era possível usá-las para todas as operações com números comuns e os 5 códigos não utilizados dos valores que não existiam foram adotados para indicar as operações! (figura 13)

O leitor também perceberá que usando representações desta forma, operavam os primeiros computadores, apropriadamente chamados de computadores de “4 bits”.

Outros Códigos

Outros códigos binários, mas não tão importantes neste momento, são o Código Biquinário, em que cada dígito tem um peso e são sempre usados 7 bits para sua representação e o Código Gray que aparece em diversas versões.

O Código Gray se caracteriza pelo fato da passagem de qualquer número para o seguinte sempre ser feita com a mudança de um único dígito.

Assim, por exemplo, quando passamos de 0111 (7 em decimal) para 1000 (8 em decimal) os quatro dígitos mudam. No Código Gray a passagem do 7 para 8 muda apenas um dígito, pois o 7 é 0100 e o 8 é 1100.

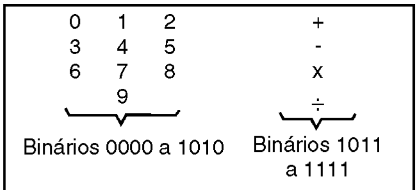

Figura 13 - Uso dos valores de 0000 a 1111.

Podemos ainda citar os Códigos de Paridade de Bit e o Código de Excesso 3 (XS3) encontrados em aplicações envolvendo circuitos digitais.

1.7 - SISTEMA HEXADECIMAL

Os bits dos computadores são agrupados em conjuntos de 4, assim temos os computadores de 4, 8, 16 e 32 bits. Também observamos que com 4 bits podemos obter representações binárias de 16 números e não somente de 10. Vimos que os 5 excedentes poderiam ser usados para representar operações nas calculadoras.

Isso significa que a representação de valores no sistema hexadecimal ou de base 16 é mais compatível com a numeração binária ou operação binária dos computadores.

E de fato isso é feito: abrindo muitos programas de um computador, vemos que suas características como posições de memória ou quantidade de memória são feitas neste sistema.

Isso significa que o técnico precisa conhecer este sistema e mais do que isso, deve saber como fazer conversões dele para o decimal e vice-versa, além de conversões para o sistema binário. Na tabela abaixo damos as representações dos dígitos deste sistema com equivalentes decimais e binários:

Decimal	Binário	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Observe que como não existem símbolos para os dígitos 10, 11, 12, 13, 14 e 15, foram usadas as letras A,B,C,D,E e F.

Como fazer as conversões: os mesmos procedimentos que vimos para o caso das conversões de decimal para binário e vice-versa são válidos para o caso dos hexadecimais, mudando-se apenas a base.

Vamos dar exemplos:

Como converter 4D5 em decimal:

Os pesos no caso são: 256, 16 e 1. (a cada dígito para a esquerda multiplicamos o peso do anterior por 16 para obter novo peso).

Temos então:

4D5 = (4 x 256)+(13x16)+(1x5) = 1237

Observe que o “D” corresponde ao 13. O número decimal equivalente ao 4D5 hexadecimal ou “hex”, como é muitas vezes representado, é 1237.

4D5 (hex) = 1237 (dec)

A conversão inversa, ou seja, de decimal para hexadecimal é feita por divisões sucessivas. Tomemos o caso de 1256, apresentado na figura 14.

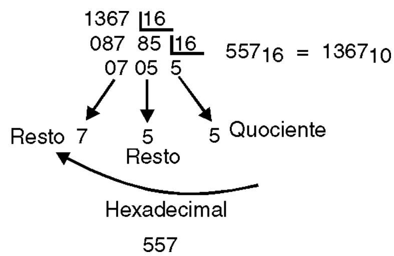

Figura 14 - 1367 decimal equivale a 557 na base 16.

Veja que basta ler o quociente final e depois os restos das divisões sucessivas, sempre lembrando que os que excederem 10 devem ser “trocados” pelas letras equivalentes.

EXERCÍCIOS

a) Converter 645 em BCD
b) Converter 45 em binário puro
c) Converter 11001 (binário) em decimal
d) Converter 1101 0011 1011 (BCD) em decimal
e) Converter 1745 (decimal) em hexadecimal.
f) Converter FFF (hex) em decimal.
g) Converter F4D (hex) em decimal.